Computer Networks 79 (2015) 236–246

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Impact of device unavailability on the reliability of multicast transport in IEEE 802.11 networks

Yousri Daldoul [a,b], Djamal-Eddine Meddour [a,*], Toufik Ahmed [b], Raouf Boutaba [c]

[a] Orange Labs Network, Lannion, France
[b] Bordeaux I University, Bordeaux, France
[c] School of Computer Science, University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 3 April 2014
Received in revised form 5 December 2014
Accepted 16 January 2015
Available online 28 January 2015

Keywords:
IEEE 802.11
Reliable multicast
Loss diagnosis
Device unavailability

ABSTRACT

Multicast transport is an efficient solution to deliver the same content simultaneously to many receivers. This transport mode is mainly used these days to deliver real-time video streams. However, multicast transmissions support over IEEE 802.11 networks does not provide any feedback policies, which implies a definite loss of missing packets. This impacts the reliability of the multicast transport and the application employing it. An alternative to improve the reliability of multicast streaming over 802.11 networks is to prevent packet losses. In this perspective, it is necessary to identify the loss causes and to perform the required prevention actions. It is well known that collisions and path loss are two fundamental sources of transmission failures. Their impact can be eliminated by means of collision prevention and data rate adaptation. However, several works show that the loss rate of multicast packets may be considerable even in collisions-free environments and using an appropriate transmission rate. Particularly they show that losses may have a bursty nature which does not correspond to the bit error rate model of the PHY layer as defined by the chipset manufacturers. Therefore, in this paper, we carry out a thorough investigation of the loss causes in wireless networks. We show that device unavailability may be the principal cause of the significant packet losses that occur and their bursty nature. Particularly, our results show that the CPU overload may incur a loss rate of 100%, and that the delivery ratio may be limited to 35% when the device is in the power save mode.

1. Introduction

Packet losses in a WLAN may occur due to many factors. These losses reduce the network throughput, impact the reliability of the wireless link, and increase the transmission delays. This is because missing packets are considered as wasted transmission time, and retry-based protocols require additional delays to send data again. Besides, the losses reduce the delivery ratio of multicast and even unicast transmissions. Therefore, it is necessary to identify the factors that cause losses and perform the required actions so as to avoid transmission failures. This will enhance the reliability and the latency of multicast transport, and will improve network throughput. Interferences (including collisions) and path loss are two fundamental loss factors in wireless networks. Efficient mechanisms exist and are used to eliminate their impact. For instance, interferences are avoided by means of good planning of the available channels. Multicast packets are typically protected against collisions using CTS-to-Self [1] or Busy Symbol [2]. On the other hand, several rate adaptation schemes have been defined to select the appropriate transmission rate for multicast flows [3,4] to control the path-loss effect.

* Corresponding author. Tel.: +33 296 072 936; fax: +33 296 071 852.
E-mail address: djamal.meddour@orange.com (D.-E. Meddour).

http://dx.doi.org/10.1016/j.comnet.2015.01.008
1389-1286/

However, many studies show that the loss rate can be significant even in the absence of interferences and when using an appropriate transmission rate. They also show that losses may occur in bursts. This loss pattern does not fit with the bit error rate model of the PHY layer as defined by the chipset manufacturers (see Figs. 14 and 15 of [5] for examples). Based on our hypothesis that transmission failures should occur individually and randomly, and our observation that the loss rate sometimes depends on the device itself, we argue that the device performance may be the main cause of the significant yet unexplained losses and their burstiness nature. It is worth noting that if losses are bursty due to the wireless channel, even the unicast transport becomes unreliable. This is because a packet and its retransmissions may be lost within a burst. This casts doubt on the reliability of the 802.11 unicast transport.

In this paper we carry out a comprehensive analysis of the loss factors in wireless networks using an experimental test-bed. We particularly focus on the case of multicast transport. We identify a new loss factor termed device unavailability. We show that the device is unable to process any packet when: (1) it is sleeping according to the Power Save (PS) mode; and (2) the Control Processing Unit (CPU) is overloaded with tasks exceeding the processing capability of the device. As such, we show that the device is itself responsible for missing several multicast packets and that the loss rate due to device unavailability may reach 100%. We highlight that this factor is widely ignored particularly in the field of wireless communications where mobile devices have limited battery and CPU resources. Furthermore, we provide recommendations for avoiding such losses and thereby enhancing the reliability of multicast transmissions.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the PS mode and the evaluation of its impact on the delivery ratio and throughput of multicast flows. We introduce the CPU overflow issue and its impact on loss rate in Section 4. Finally, we conclude in Section 5.

2. Related works

There has been intensive work on analyzing the different causes of packet loss in wireless networks [6–24]. The underlying motivation is to understand the reliability of the wireless link and the variation of available bandwidth. Another motivation is to define rate adaptation algorithms at the sender that are aware of the loss sources and capable to decrease the transmission rate when necessary in order to maintain a reliable communication link with the receiver.

Most existing data rate adaptation algorithms [25–34] consider that losses in a WLAN occur either due to radio signal deterioration or due to collisions. The typical reaction to a reduced Signal-to-Noise-Ratio (SNR) is to decrease the transmission rate since a lower rate is always more robust. But if the rate adaptation algorithm believes that the transmission failure is caused by simultaneous channel accesses, it does not reduce the data rate and let the MAC layer increase the contention window in order to reduce the collision probability at the following transmission attempt.

In [6] the authors evaluated the impact of multipath fading and interferences on the loss rate of multicast packets. They deployed two networks operating at neighboring channels (namely, channels 10 and 11) in the same room. They showed that the loss rate varies between 2% and 7% in the presence of interferences from channel 10. Then they disabled the interfering network and they evaluated the impact of multipath on packet losses. They considered the scenario of a moving person in order to create reflections and showed that the average loss rate in this scenario is about 0.3%. Finally, they showed that the average rate falls to 0.1% in a static environment. The authors concluded that this variation is related to the multipath effect. However, we believe that a variation of 0.2% obtained using randomly chosen devices is not necessary enough to make important conclusions. Therefore it is necessary to prove that measurements are achieved using dedicated and precise equipment. Otherwise, the variation may be simply related to device unavailability. Therefore, the authors of [6] should show that the loss rate is not affected by the device unavailability, and should illustrate the variation of this rate over time to prove whether or not a variation of 0.2% is trustworthy and depicts the multipath effect.

In [7] the authors studied the correlation of packet losses as experienced by the receivers of multicast/broadcast streams. This is typically referred to as the spatial loss correlation. In their paper, the authors showed that closely located receivers loose the same packets almost all the time when the received signal is strong enough. This correlation decreases and the losses become independent when the radio signal deteriorates. In [34], the authors evaluated the loss rate as a function of the signal fluctuation. They introduced a new model to estimate the bit error rate of the wireless link based on the notion of effective SNR. Another experimental study was conducted in [8] where the loss rate of the multicast transport is measured, and the unfair sharing of the medium between unicast and multicast flows is evaluated. In this study, authors classified losses into three categories: "collisions", "queue overflow" and "others". However, they did not investigate the causes of transmission failures in the case of the third category. In [9] the authors collected loss traces from a real 802.11b network and studied their spatial correlation. They also evaluated the burstiness nature of the missing packets called temporal loss correlation. They showed that the loss rate varies between 4% and 30% from one receiver to another. In their testbed, the station with the highest loss rate (i.e., 30%) is the nearest one to the AP. This suggests that the device with the best link quality may experience the highest loss rate. Our work shows that device unavailability (PS mode and CPU overflow) may increase the loss rate significantly. Therefore, a receiver with good link quality but with bad configuration (in PS mode or with overloaded CPU) may experience high loss rate. We believe that this is the case in [9], where the device with the best link quality experiences the highest loss rate. The same observation is made in [10] where the authors confirm that identical devices perform quite differently in terms of packet error rates. This conclusion is reached using a multicast group of 9 collocated mobile phones and an IEEE 802.11b network. Specifically, the authors showed that the loss rate varies between 4% and 94% from one receiver to another.

Several hypotheses regarding the causes of transmission failures are explored in [11]. The authors showed that the loss rate depends on the signal attenuation and interferences. They also investigated the losses caused by the multipath effect showing that the loss rate increases when the delay between the direct and the reflected signals is higher than several hundreds of nanosecond. They found that some nodes among the 38 deployed stations experience important and frequent bursty losses. Moreover, these bursts and their frequency neither depend on the transmission rate nor on the distance between the sender and the receiver. However, the authors do not provide any explanation for this loss pattern. In [12] the authors showed that a missing burst can be as long as several hundred packets, but the size of loss free bursts may exceed 10,000 packets. Several other works reached similar conclusions [13–15], i.e., those losses may occur frequently in bursts. However, there has been no explanation that establishes with certainty why this pattern occurs and why it does not necessary affect all receivers. The authors in [14] speculated that synchronization failures are possible causes of these losses. We do not share this analysis. Indeed, the IEEE 802.11 defines a communication system capable to ensure a very high success rate for data symbols that the receiver ignores before their successful reception. Besides, the synchronization symbols are known a priori and are transmitted using a dedicated modulation scheme which is more robust than those used to carry the data. As such one can expect the success rate of the synchronization symbols reception to be significantly higher than that of data symbols. It is worth noting that all existing simulators consider that the synchronization success rate is always 100%.

In summary, none of the aforementioned studies investigated the issue of device unavailability, which we believe, as we will show in this paper, has a significant impact in terms of losses, and is therefore necessary to take into account in order to define appropriate prevention actions and to improve network efficiency.

3. Investigating the power saving mode

3.1. Operating mode

The IEEE 802.11 introduced the Power Save (PS) mode in order to reduce the energy consumption of the connected devices. In a Base Service Set (BSS), the Access Point (AP) sends a Beacon frame periodically. This control packet contains the Traffic Indication Map (TIM) which identifies the Stations (STAs) for which traffic is pending and buffered at the AP. It also notifies the presence of multicast/broadcast packets. Two different TIM types are defined: TIM and Delivery TIM (DTIM). Every specific period of DTIMPeriod, a TIM of type DTIM is transmitted within the Beacon frame rather than an ordinary TIM.

If any STA in the BSS is in PS mode, the AP should buffer all multicast and broadcast packets, and deliver them immediately following the next Beacon including a DTIM. The MoreData field is set in the headers of all packets except the last one to indicate the presence of further buffered broadcast/multicast packets. These packets should be sent before transmitting any unicast traffic.

A STA that stays awake to receive broadcast/multicast traffic should remain awake until (1) the MoreData field of the broadcast/multicast packets indicates there is no further buffered broadcast/multicast traffic, or (2) a TIM is received indicating there are no more buffered broadcast/multicast packets.

The current power management of the standard allows the implementation of several standard-compliant energy saving levels. Depending on the power management requirements of a STA, this latter may choose to wake up every TIM, DTIM or else. A STA may have its ReceiveDTIMs variable set to “False” which means that the STA is not required to wake up every DTIM. This configuration may be selected to further reduce power consumption, but leads to frequent multicast packet losses.

Fig. 1 illustrates the AP and STA activity under the assumption that a DTIM is transmitted once every 3 TIMs. The first line represents the AP activity. The AP schedules Beacon frames for transmission every Beacon interval. The second and third lines depict the activity of two STAs operating with different power management requirements. In this example, the second STA has ReceiveDTIMs set to “False” and does not wake up at every DTIM. Thus, STA2 may miss several multicast packets.

The first STA, powers up its receiver and receives a TIM in the first Beacon frame; that TIM indicates the presence of a buffered packet for that STA. STA1 then generates a PS-Poll frame, which elicits the transmission of the buffered packet from the AP.

The Second Beacon contains a DTIM. Hence the AP sends all the buffered multicast packets. STA1 remains awake to receive these packets. However, in our example, STA1 misses the last packet. Hence it remains awake until the third Beacon which indicates that there are no more multicast/broadcast packets.

STA2 wakes up before the fourth Beacon in order to receive it correctly. As the TIM indicates buffered packets for STA2, this latter sends a PS-Poll to the AP which sends the buffered packet. At the fifth Beacon interval, the AP sends a Beacon including a DTIM field. This field indicates the availability of multicast packets in addition to unicast packets for STA1. After receiving the last multicast packet, STA1 sends a PS-Poll to the AP which sends the buffered unicast packet.

Since the default Beacon interval is 100 ms, if the DTIMPeriod value is set to 3, multicast services may experience latency in the order of 300 ms. This delay may impact considerably the experienced quality of real-time multicast services. Moreover, since multicast packets are transmitted immediately following a DTIM and before transmitting any unicast packet, high throughput multicast services impact considerably the latency experienced by other time sensitive unicast traffic. This impact has been evaluated in [35]. Besides, several STAs may be required to awake in

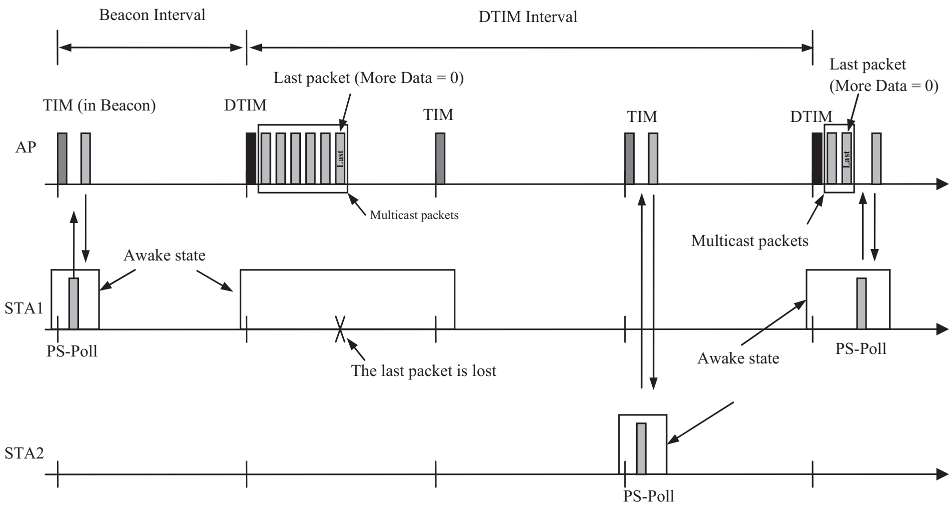

Fig. 1. Infrastructure power management operation.

order to receive multicast traffic (ReceiveDTIMs = False), even if they are not concerned by this data. Therefore, the multicast transmissions lead to unnecessary increase in power consumption [36].

Furthermore, the PS operating mode may increase the loss rate of multicast packets; these packets are transmitted without the request of the STAs which may be sleeping and may therefore miss all the transmissions. Also, the PS mode may impact the throughput of multicast services; by limiting the multicast transmissions to specific time intervals, the available bandwidth for multicast traffic will be reduced. We emphasize that this impact may be experienced even if all the STAs in the PS mode are not members of the multicast group and are not interested in the multicast traffic. This is because the AP is not aware of group memberships.

Table 1
Configuration parameters.

Parameters	Values
Network	802.11a
Beacon interval	100 ms
DTIMPeriod (resp. DTIM interval)	1 (resp. 100 ms)
CWmin	31
Atheros chipsets (AP)	Atheros AR9300-XB112
Intel card (STA1)	Intel® WiFi Link 5100 AGN
Atheros chipsets (STA2)	Atheros AR9300-XB112
Computer AP	Dell Latitude D420
Driver AP	Ath9k (Atheros)
Computer STA1	Dell Latitude E6400
Driver STA1	NETw5 (Intel)
Computer STA2	Dell Latitude D420
Driver STA2	Ath9k (Atheros)

3.2. Impact on reliability and throughput

In the remainder of this section we evaluate the impact of the PS mode on reliability and throughput of multicast transport. We conduct our experiments in a closed area without or with very limited interferences with other systems. We consider a WLAN composed of one Access Point (AP) and two stations: STA1 and STA2. The characteristics of the network and the devices are given in Table 1. All three laptops are powerful Dell computers that use recent 802.11n chipsets. We install the AP and the two STAs close to each other in order to avoid failures caused by path loss.

In our evaluations, we notice that the default configuration of the AP driver does not block forwarding multicast packets from the AP to STA1/STA2, when any of the associated STAs is in the PS mode. In fact, these packets are stored in a special queue (PS queue) at the MAC layer, and are transmitted after the DTIM. The PS queue is used together with the primary queue. However, the MAC layer notifies the network stack only about the saturation state of the primary queue. Thus when the PS queue is full, new arriving multicast packets are automatically rejected. In order to accurately evaluate the losses caused by the PS mode, we only consider the effectively transmitted multicast packets. Hence we collect statistics directly from the transmitter module of the AP driver.

The two chipsets used in the experiments provide different power management levels. The Intel card can operate at 6 different levels of power savings numbered 0 through 5 [37]. Level 0 disables the PS mode (i.e., the device is fully powered) and level 5 provides the greatest amount of savings. The Atheros chipset provides only two levels: ON (device in PS mode) and OFF (no PS). We measure the packet delivery ratio for a video of 25 frames per second (fps) and an average bitrate of 1360 Kbps. The multicast application is running on the AP and delivers this video using RTP. Accordingly, one image is packetized and forwarded to the MAC every 40 ms.

As depicted in Table 1, DTIM interval is set to 100 ms. Only the AP defines the values of TIM/DTIM. Any associated station should operate accordingly. On the other hand, associated stations are able to determine the value of DTIM by reading the beacon frame. However, a station may choose to wake up at every DTIM or not. Unfortunately, we were not able to determine how the PS is implemented for Intel because the driver is not open source. Nevertheless, we showed that Intel implements 6 power save levels. On the other hand, ath9k driver is open source, but the PS mode is implemented at two levels: (1) ath9k driver and (2) hardware. The hardware part of the implementation is not accessible. However, the driver part is implemented according to the following flowchart (Fig. 2).

Fig. 3 illustrates the reception ratio of STA1 for the different power levels. For each level, we transmit the same video 10 times. In each of these experiments we measure the packet delivery ratio. Our results show that the loss rate depends on the power level. Specifically, for levels 5 and 4 the delivery ratio varies between 33% and 63%. This ratio increases at levels 3, 2 and 1 and reaches 90%. The loss ratio becomes less important at level 0 where the delivery ratio is about 98%.

We also evaluate the reception performance of STA2 for the 2 supported levels: power management ON and OFF. We compare the delivery ratio of STA1 and STA2 in Fig. 4. During the 10 first trials, we set STA1 at level 2 and STA2 at level ON. We switch to levels 0 and OFF respectively for the 10 last trials. We observe that the Atheros chipset outperforms the Intel one. Moreover, we notice that the delivery ratio in a collision free medium is higher than 99% for STA2 when it is fully powered.

In the remainder of this section we evaluate the impact of the PS mode on the throughput of multicast services in a real network. We consider a Constant Bit Rate (CBR) stream and measure the number of transmitted multicast packets for the following transmission data rates: 6 Mbps, 12 Mbps, 24 Mbps and 54 Mbps. We increase the packet length progressively, and we compare the average transmitted packets per second during the PS operation and in a fully powered (FP) network. We compare the FP rate with the theoretical one. We illustrate the obtained results in Fig. 5.

In Fig. 5(a), all packets are transmitted at 6 Mbps. We observe that the transmission rate under PS operation is about 500 packets per second (pps) and is bounded by the rate of the fully powered network. Also we notice that

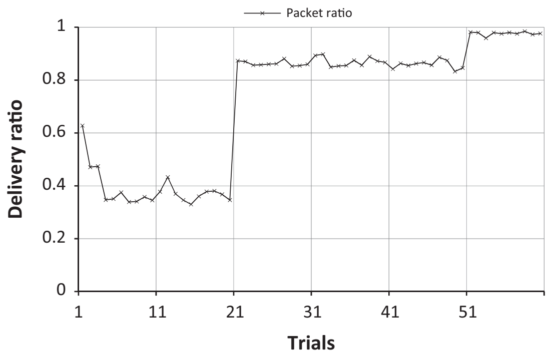

Fig. 3. Impact of PS mode on the packet delivery ratio – STA 1.

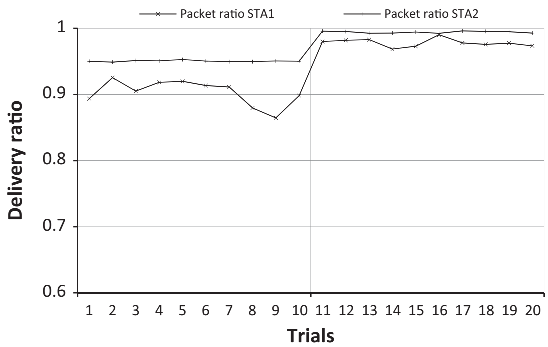

Fig. 4. Impact of PS mode on the packet delivery ratio – STA 1 vs STA1.

the theoretical results are higher than those of the experimental testbed. We speculate that this is due to the device performance. As we will show in the next section, the device may be unavailable during some instants, and unable to transmit. Hence the bandwidth is not totally used.

We observe similar behavior as shown by the curves in Fig. 5(b)–(d). Multicast throughput under the PS operation is about 500 pps regardless of the packet length and of the used PHY data rate. On the other hand, the theoretical

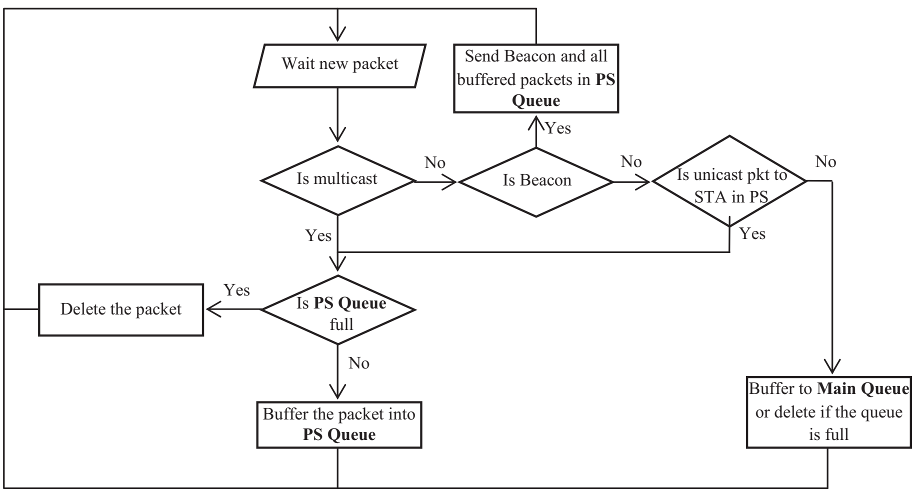

Fig. 2. Flowchart of Ath9k packet transmission procedure in PS mode.

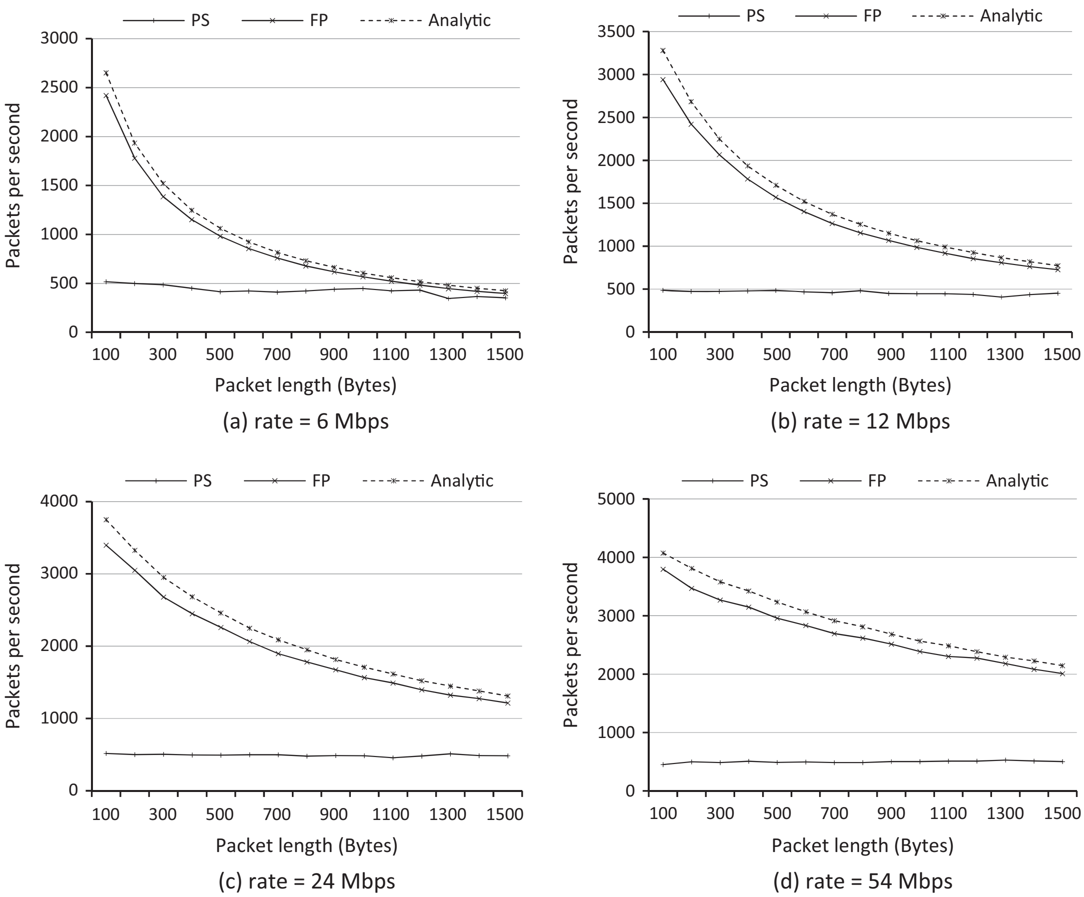

Fig. 5. Multicast packets transmission rate for PHY rates of 6 Mbps, 12 Mbps, 24 Mbps and 54 Mbps.

throughput slightly exceeds the throughput of the fully powered network.

An obvious solution to avoid the high loss rate caused by the power save mode is to disable it. It may be disabled temporarily during the streaming duration using Groupcast with Retries – Active method (GCR-A) [38]. This delivery method awakens sleeping receivers, and allows the AP to send multicast packets at any time, regardless of the power state of the group members. The primary goal of designing the PS mode is to reduce power consumption of devices that are idle most of the time. However, multimedia services deliver high throughput traffic requiring the devices to be continuously listening.

However, it is necessary to distinguish between multicast members that should be fully powered, and non-members that may be allowed to stay in the PS mode. Therefore, a reliable and energy-efficient multicast protocol should be aware of every group member allowing a receiver to switch back to the PS mode once it is not a member of the multicast service anymore.

3.3. Analysis of PS mode operations

We showed that the PS mode of IEEE 802.11 does not necessary lead to packet losses; as described in Section 3.1, a receiver which is configured to wake up at each DTIM interval (such as STA1 in Fig. 1) should be able to receive the multicast packets without significant losses. Only stations in deep PS mode (i.e. those that are not required to wake up at each DTIM, such as STA2 in Fig. 1) may experience very important loss rates. However, according to Fig. 3, we suspect ReceiveDTIMs to be false when Intel chipset is in levels 5 and 4, and to be true when the chipset is in levels 3, 2 and 1. Unfortunately, we cannot prove this assumption because we do not have access to the driver's source code. On the other hand, when Ath9k is in PS mode, the value of ReceiveDTIMs is true. Therefore the device is supposed to wake up at each DTIM interval. However, our results show that even through ReceiveDTIMs = true, the receiver experiences a high loss rate of about 10% for Intel and 5% for Atheros (as demonstrated in Fig. 4). We deduce that the PS mode of these devices is not compliant with the standard requirements. We note that a device does not need to be compliant with all the functionalities of 802.11 in order to be available in the market. However, a WiFi certification is delivered to equipments that are compliant with a well-defined subset of functionalities. As an example, devices certified "*WiFi-n*" do not need to support PHY layers other than 802.11n. Therefore, we recommend the integration of PS functionalities within new certifications in order to enhance the reliability of the multicast transmissions during PS mode. Furthermore, Fig. 5

shows that the throughput of multicast services is significantly reduced when the device is in PS mode. However, multicast is mainly used to deliver high throughput traffic such as IPTV/HDTV. Therefore the current PS mode should be enhanced in order to satisfy all the QoS requirements. We recommend the use of a new PS scheme that operates as follows:

– When a high throughput service is delivered, the PS mode is disabled during the session duration.
– This mode is enabled again when the session is over.

This new scheme increases the availability of the receiver during the streaming period and allows the user to enjoy high throughput services. When the session is over, the device enables the legacy PS mode in order to reduce the power consumption and to improve the battery lifetime.

4. Investigating the CPU overflow

End user terminals such as computers and smartphones are composed of a Central Processing Unit (CPU) that is typically connected to several other devices including network chipsets, screen, hard disk, storage devices, CD reader, sound card and keyboard. The CPU is the host of the operating system which manages all the plugged devices. The latter are themselves equipped with less sophisticated but dedicated processors called microprocessors/microcontrollers.

Hardware interrupts constitute the principal method for communication between the CPU and the attached devices, including the 802.11 chipsets. Indeed, when a new packet is received from the wireless medium, the microcontroller puts the data in a shared buffer and triggers an interrupt. Under the assumption that the CPU is available, the packet is processed and forwarded to the appropriate application. The way an interrupt is handled depends on the operating system and may evolve from one release to another. Table 2 illustrates some potential sources of interrupts according to [39].

Note that a processor may handle one hardware interrupt at a time. If a new interrupt occurs from another interface while the processor is still handling a hardware interrupt, the new one is lost. If the rate of interrupts occurrence exceeds the processing capacity, several tasks will be delayed or even discarded. For the case of 802.11 chipsets, several packets (including those received correctly) will be dropped. The impact of these drops is similar to packet losses from the application point of view. Besides, they impact the reliability of both unicast and multicast transmissions. It is worth noting that the impact of the interrupts on the CPU depends on the amount of work required to process these interrupts. For example an interrupt coming from the keyboard when a key is pressed requires less processing time than the required time to process a large packet coming from the network device with many parameters such as the reception power, data rate, sender and receiver addresses, etc.

Table 2
Potential sources of excessive interrupts for embedded processors.

Source	Interruption rate (per second)
Serial port @115 kbps	11,500
10 Mbps Ethernet	14,880
CAN bus	15,000
I2C bus	50,000
USB	90,000
100 Mbps Ethernet	148,800
Gigabit Ethernet	1,488,000

4.1. Impact on the delivery ratio

In the remainder of this section we measure the impact of packet arrival rate on packet loss rate as experienced by multicast applications. We build an IEEE 802.11a network on a free channel in order to avoid collisions and interferences. We use AP and STA1 of Table 1 as the sender and the receiver, respectively. We configure the sender to use CWmin = CWmax = 0. Therefore packets are separated with a DIFS time period. Moreover, all multicast packets have 100 byte-length (including the MAC header) and are delivered at 54 Mbps. This scenario is designed to deliver packets with a very high rate and to incur an important interrupt rate at the receiver. The two terminals are separated with less than 0.1 m. Besides, we set the transmission power of the AP to the maximum allowed level, i.e. 100 mW. On the other hand, the received signal strength at the associated station is always higher than 70 dB. By using the maximum transmission power at a very short distance, we are sure that the received signal is very high and ensures a reliable communication at all the data rates. As such, any packet loss is caused by the device performance (i.e., CPU unavailability or chipset failures). We increase the packet transmission rate progressively and we record the delivery ratio. For each rate, we perform between 10 and 20 measurements. In each measurement we send 100,000 packets at a constant rate. We make only 10 measurements for the low packet rates (rates between 1000 pps and 8000 pps) because the delivery ratio appears to be stable. Then we increase the number of measurements starting from the rate of 9000 pps. As such, we perform up to 20 measurements when the rate varies between 9000 pps and 14,000 pps. We show the obtained results in Fig. 5. We note that the maximum achieved rate is limited to 12500 pps. Thus the results illustrated for throughputs of 13,000 pps and 14,000 pps are obtained for an effective packet transmission rate of 12500 pps. In fact, in our testbed we consider 3 different levels of measurement: (1) application level at the sender, (2) MAC level at the sender and (3) application level at the receiver. We vary the packet buffering rate of the first level from 1000 pps to 14,000 pps. This rate is illustrated in Fig. 6. However, based on our measurements, the maximum transmission rate of the MAC layer (i.e the second measurement level) is limited to 12,500 pps. This means that when the buffering rate is lower than 12,500 pps, all the buffered packets are transmitted. Beyond this value, the transmission rate is limited to 12,500 pps and several packets are dropped due to the queue overflow. We exclude the dropped packets as they are not transmitted over the wireless medium, and there-

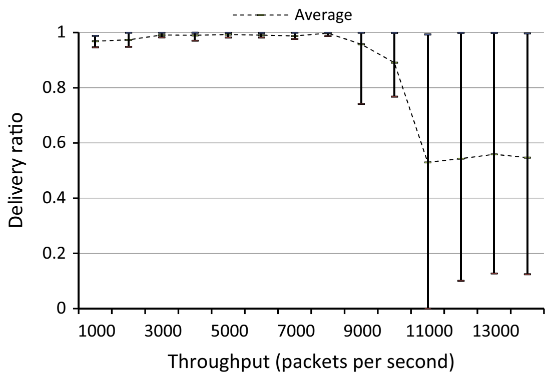

Fig. 6. Impact of the packet transmission rate on the delivery ratio of multicast packets.

fore, they do not affect the CPU load of the receiver. So we determine the delivery ratio by comparing the number of the received packets with that of the effectively transmitted ones (i.e. the MAC level sent packets). As the CPUs of both the sender and the receiver have different characteristics and load levels, the receiver saturates starting from 11,000 pps although the sender is able to reach 12,500 pps.

Fig. 6 shows the maximum, the minimum and the average delivery ratios using the results of the different measurements. We observe that the average ratio is about 98% for relatively low rates of 1000 pps and 2000 pps. For these rates, the inter-arrival periods are the longest. Then the average delivery ratio increases and varies between 99.1% and 99.8% for rates from 3000 pps to 8000 pps. This let us deduce one or both of the following two postulations: (1) the CPU attributes more attention to devices triggering frequent interrupts; and/or (2) the network device is more ready to receive when the packet arrival rate increases. We unfortunately cannot provide the exact reason of this slight variation, but we think that this is related to the design of the operating system. In fact, it is well known that the processor loads frequently used tasks to high-speed access areas for optimization purposes. We deduce that there is a similar optimization that allows the processor to handle more effectively a hardware triggering frequent interrupts. Such optimization would enhance the delivery rate when the packet arrival rate increases (e.g. from 1000 pps to 3000 pps).

Further, we notice that the delivery ratios vary between 99.9% and 74.1%, and the averages are 95.8% and 89.1% for packet rates of 9000 pps and 10,000 pps, respectively. Then we observe a very high loss rate reaching 100% starting from 11,000 pps. At this level, the average delivery ratio is limited to 56%, although the loss rate is sometimes very low (less than 0.3%). We note that the case of 0% delivery ratio illustrates one of the measurements where the application does not receive any packet from the 100,000 transmitted ones. During this experiment, the computer hangs and does not respond to any event till the end of the multicast session. Then, it returns to ordinary operations without needing a reboot. For all the other tests, the computer responds to any request but a slight slowdown is observed when we move the mouse. This is observed when the packet rate increases (starting from 11,000 pps).

As the multicast packets are separated with DIFS (recall that CW = 0), the theoretical throughput is obtained according to Equation below. We note that 10 beacons are sent each second at a rate of 6 Mbps. Therefore, their transmission duration should be removed in order to obtain the remaining time for sending the multicast packets.

$$\mathrm{Npps} = (1 - 10 \times (\mathrm{PIFS} + \mathrm{Beacon\_duration}))/(\mathrm{DIFS} + \mathrm{T_{PPDU}})$$

The TPPDU is the packet transmission duration which depends on the packet size and the used PHY data rate. Besides, the Beacons sent on our testbed are 90 bytes size and with a transmission duration of 169 μs at 6 Mbps. Finally, PIFS and DIFS are standard spaces having durations of 25 μs and 34 μs, respectively.

As we mentioned above, the highest achieved rate is 12500 pps. However, the expected rate is 14261 pps. Thus the achieved throughput is bounded by the CPU capacity of the sender. To confirm this finding, we consider a second scenario of larger packets of 1500 byte-length transmitted at 54 Mbps and 6 Mbps, and we compare the expected and the achieved throughputs. These results are illustrated in Table 3. In fact, the transmission duration increases when: (1) the packet size increases from 100 to 1500 bytes (PHY rate stays at 54 Mbps); and (2) the PHY rate decreases from 54 Mbps to 6 Mbps (packet length stays at 1500 bytes). Therefore the transmission rate (expressed in pps) decreases. This reduces the load, enhances the performance of the CPU at the sender, and limits therefore the gap between theoretical and achievable throughput. As such, Table 3 confirms that the achieved throughput is impacted by the CPU capacity.

Moreover, we observe that the delivery ratios using 6 Mbps and 54 Mbps are almost the same although these data rates have different robustness degrees against bit errors. This confirms that the losses are not caused by signal distortion but they are rather due to receiver unavailability. Furthermore, we notice that even the reliability of the unicast transport is considerably impacted by the CPU unavailability. However, the delivery ratio of the unicast outperforms that of the multicast when the packet transmission rate is low. This is because losses occur individually and the packets are received correctly after a retransmission attempt. But when the packet rate increases, losses occur in bursts of important sizes. Therefore a unicast packet and its retransmissions are all lost. This leads to packet rejections. To deal with these losses, one or both of the following solutions should be considered: (1) The actual rate adaptation algorithms should take into account the fact that a packet may be lost due to CPU overload; (2) The CPU capacity should be enough to process any arriving packet.

To obtain these results we use a powerful computer with two processors and no active acquisition device other than the 802.11 chipset (the keyboard is kept idle while the mouse is softly used). Hence, a less sophisticated receiver with one single processor (such as a smartphone or a tablet) may experience important losses starting from lower packet reception rates. Therefore, the use of a terminal with an appropriate configuration is the first require-

Table 3
Impact of the CPU performance on the packet transmission rate and the delivery ratio in a highly loaded network (i.e. the transmission queue of the sender is never empty).

Mode	Packet length (bytes)	PHY rate	Highest theoretical rate (pps)	Highest achieved rate (pps)	Gap (%)	Delivery ratio (%)
Multicast	100	54 Mbps	14,261	12,500	12.35	12.45–99.73
	1500	54 Mbps	3591	3333	7.18	99.88
	1500	6 Mbps	485	471	2.89	99.69
Unicast	100	Ath9k rate adaptation	–	–	–	11.2–99.99

ment in order to take advantage of a multicast session with a high quality, but also to take a full advantage of the very high throughput capability of future networks.

It is worth noting that any coming packet (including that arriving with PHY or MAC errors and that addressed to another receiver) generates an interrupt and increases the processor load. Furthermore, disabling the promiscuous mode does not resolve the issue. This is because this mode is implemented by the device driver, and filters the packets already processed by the CPU. Therefore it is necessary to implement an alternative promiscuous mode at the network chipset itself in order to get ride of useless interrupts. This is required particularly with the imminent arrival of very high throughput networks capable of transmitting a huge number of packets to different receivers. The stations with limited processing capabilities will be even more impacted as they will not be able to handle such a big number of packets.

4.2. Recommendations for new rate adaptation schemes

Rate adaptation algorithms can play a very important role in estimating the loss factors and in determining suitable actions. Considering that the failure is caused due to link deterioration, rate adaptation selects a more robust data rate. Such decision allows the communication to occur reliably. On the other hand, considering that a failure is due to a collision, the appropriate action is to increase the Contention Window in order to reduce the collision probability. Another solution to avoid the collision is to enable the use of RTS/CTS temporarily. However, existing rate adaptation schemes do not take into account the fact that a failure may occur due to CPU overflow. Therefore, they do not deal with these failures appropriately.

A suitable approach to deal with the case of CPU overflow is to delay the transmission. Therefore, the sender needs to buffer the packets in a dedicated queue (similarly to packets delivered in PS mode). Once the defined buffering delay expires, the packet is retransmitted. This policy reduces the packet transmission rate to the overloaded receiver and limits the losses. It is worth noting that buffering the packets of the overloaded receiver in a secondary queue is necessary in order to allow the sender to deliver other packets to other receivers. Furthermore, we highlight the fact that Backoff timer is defined to reduce the collision probability, and is limited to several μs. This delay is not enough to deal with the CPU overflow. We believe that a suitable buffering delay depends on the history of packet losses of each receiver. Therefore, we believe that a dedicated study is required to devise a suitable rate adaptation scheme that addresses the CPU overflow issue.

4.3. System failure

Like any other electronic device, 802.11 terminals may experience internal system failures and may require a reboot. We highlight that this reboot is limited to the chipset (more precisely to the micro-controller) and does not concern the host terminal (e.g., the PC or the smartphone). Besides, the reboot is performed by a component of the chipset, conventionally called watchdog, and therefore does not require a human intervention. The aforementioned failures reduce the reception availability of the device and lead to packet losses. However, they are very limited and incur a loss rate lower than 0.1% [40]. We are not able to determine the exact value of this rate for the case of 802.11 chipsets because it is not possible to separate system failures from the losses caused by CPU unavailability. But we agree that this rate is much lower than 0.1%. This is because during our empirical measurements, the device was able to receive more than 99.957% of the multicast packets. If we consider that system failures occur with a constant rate, in this case this rate is less than 0.05%.

5. Results exploitation

The principal objective of our work is to highlight the phenomenon of device unavailability that exists, has significant impact, but has not been identified in prior art. Although the results depend on the used hardware, the unavailability issue may occur at any device and may affect the accuracy of any testbed results. In fact, ignoring the impact of this issue, may lead to wrong conclusions.

We note that we obtained these results using powerful laptops and relatively low throughput networks. With the proliferation of communication devices with low CPU capabilities (e.g. Smartphones, tablets, notebooks, etc.) and with the imminent arrival of very high throughput networks (802.11ad supporting up to 6 GBps), the issue of the device unavailability may affect the performance of several existing protocols and may mislead several experimental studies.

Collisions and path-loss are well known loss factors in wireless networks. In this work we show that the device unavailability may increase the loss rate of multicast flows significantly. Therefore, Packet Error Rate (PER) becomes very limited in a typical network under the following three assumptions: (1) collisions are avoided, (2) packets are transmitted using the appropriate PHY data rate and (3) suitable devices with appropriate configuration and capabilities are used. In order to ensure this limited PER it is essential to select the appropriate data rate and to protect the multicast packets against the collisions. This protection may be guaranteed using standard features like CTS-to-Self. Moreover, several rate adaptation schemes are defined for the multicast transmissions [3,41]. They may be enhanced according to the aforementioned recommendations (Section 4.2), in order to consider the CPU overflow, and they may be used to select the appropriate data rate. Accordingly, the reliability of the multicast transmissions may be enhanced significantly using simple, yet efficient and standard compliant, recommendations. However, we believe that highlighting the device unavailability issue (i.e. PS mode and CPU overflow) will lead to the design of a new generation of smart protocols for wireless networks. In summary, we hope that the obtained results will have an impact on the design of protocols and management schemes, notably those pertaining to research investigations on the following:

– Multicast communication schemes that prevent all loss factors in order to enhance transmission reliability.
– Rate adaptation schemes that take into consideration CPU overflow as a possible cause of transmission failure.
– Power saving schemes that disable the PS mode temporarily when the receiver is a member of a high throughput multicast session (such as HDTV flows).
– Interrupt handling mechanisms that support differentiated service quality in order to provide high priority for sensitive flows.

6. Conclusions

This paper identified the device unavailability phenomena as the cause of considerable increases in the loss rate of multicast traffic. We experimentally analyzed two specific cases: (1) the device is sleeping according to the power save mode; and (2) the CPU is overloaded and is unable to process all arriving packets. Accordingly, we established the causes of bursty losses and the unexplained excessive losses that occur even in good reception conditions. These results complement our understanding of the loss factors in WLANs and most importantly shed light on some of the unexplained loss events by establishing their causes. We believe that these findings are useful to the research community notably for devising efficient rate adaptation schemes. Indeed, existing schemes decrease the transmission rate following packet losses. However, and shown in this paper, failures are not always caused by signal attenuation. Therefore, unnecessary rate decreases may occur, consequently reducing network throughput. Furthermore, our findings can be leveraged to configure the devices appropriately so as to take a full advantage of the high data rate capability of future IEEE 802.11 networks.

As a future work, we would like to investigate the provision of a quantitative framework to design smart rate adaptation algorithms that accounts for CPU overflow and device unavailability.

References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE std 802.11, 2012.
[2] Yousri Daldoul, Djamal-Eddine Meddour, Toufik Ahmed, A collision prevention mechanism for the multicast transport in IEEE 802.11 networks, in: The 18th IEEE Symposium on Computers and Communications (ISCC) 2013, 7–10 July, 2013, Split, Croatia.
[3] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: IEEE 802.11 Wireless Network Management, IEEE std 802.11v, February, 2011.
[4] Youngsam Park, Yongho Seok, Nakjung Choi, Yanghee Choi, Jean-Marie Bonnin, Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LANs, in: The 3rd IEEE Consumer Communications and Networking Conference (CCNC) 2006, 8–10 January, 2006, Las Vegas, Nevada, USA.
[5] Intersil HFA3861B Direct Sequence Spread Spectrum (DSSS) baseband processor, Data Sheet, February, 2002.
[6] Shafqat Ur Rehman, Thierry Turletti, Walid Dabbous, Multicast video streaming over WiFi networks: impact of multipath fading and interference, in: IEEE Symposium on Computers and Communications (ISCC) 2011, 28 June–1 July, 2011, Kerkyra, Greece.
[7] Zhe Wang, Mahbub Hassan, Tim Moors, A study of spatial packet loss correlation in 802.11 wireless networks, in: The 35th IEEE Conference on Local Computer Networks (LCN) 2010, 10–14 October, 2010, Denver, Colorado, USA.
[8] Diego Dujovne, Thierry Turletti, Multicast in 802.11 WLANs: an experimental study, in: the 9th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) 2006, 2–6 October, 2006, Malaga, Spain.
[9] Jerome Lacan, Tanguy Perennou, Evaluation of error control mechanisms for 802.11b multicast transmissions, in: The 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) 2006, 03–07 April, 2006, Boston, Massachusetts, USA.
[10] Janus Heide, Morten V. Pedersen, Frank H.P. Fitzek, Tatiana K. Madsen, Torben Larsen, Know your neighbour: packet loss correlation in IEEE 802.11b/g multicast, in: The 4th International Mobile Multimedia Communications Conference (MobiMedia) 2008, 7–9 July, 2008, Oulu, Finland.
[11] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, Robert Morris, Link-level measurements from an 802.11b mesh network, in: the ACM SIGCOMM 2004, 30 August–3 September, 2004, Portland, Oregon, USA.
[12] Chiping Tang, Philip K. McKinley, Modeling multicast packet losses in wireless LANs, in: The 6th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) 2003, 14–19 September, 2003, San Diego, California, USA.
[13] Andreas Willig, Adam Wolisz, Ring stability of the PROFIBUS token-passing protocol over error-prone links, IEEE Trans. Ind. Electron. 48 (5) (2001).
[14] Andreas Willig, Martin Kubisch, Christian Hoene, Adam Wolisz, Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer, IEEE Trans. Ind. Electron. 49 (6) (2002).
[15] Giao T. Nguyen, Randy H. Katz, Brian Noble, Mahadev Satyanarayanan, A trace based approach for modeling wireless channel behavior, in: The Winter Simulation Conference 1996, 8–11 December, 1996, Coronado, California, USA.
[16] Hamid R. Tafvizi, Zhe Wang, Mahbub Hassan, Salil S. Kanhere, Multipath fading effect on spatial packet loss correlation in wireless networks, in: IEEE Vehicular Technology Conference (VTC) Fall 2011, 5–8 September, 2011, San Francisco, California, USA.
[17] Mingu Cho, Hakyung Jung, Shinhaeng Oh, Ted Taekyoung Kwon, Yanghee Choi, Distinguishing collisions from low signal strength in static 802.11n wireless LANs, in: The 7th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT) 2011, 6–9 December, 2011, Tokyo, Japan.
[18] Wang Hao, Jiang Hao, A particle filter and joint likelihood ratio based error source diagnosing method for IEEE 802.11 networks, Wirel. Commun. Mobile Comput. (2013)
[19] Malik Ahmad Yar Khan, Darryl Veitch, Peeling the 802.11 onion: separating congestion from physical PER, in: the 3rd ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH) 2008, 19 September, 2008, San Francisco, California, USA.
[20] Shinuk Woo, Hwangnam Kim, Estimating link reliability in wireless networks: an empirical study and interference modeling, in: IEEE INFOCOM 2010, 14–19 March, 2010, San Diego, California, USA.

[21] Andreas Wapf, Michael R. Souryal, Measuring indoor mobile wireless link quality, in: IEEE International Conference on Communications (ICC) 2009, 14–18 June, 2009, Dresden, Germany.
[22] Michael R. Souryal, Luke Klein-Berndt, Leonard E. Miller, Nader Moayeri, Link assessment in an indoor 802.11 network, in: IEEE Wireless Communications and Networking Conference (WCNC) 2006, 3–6 April, 2006, Las Vegas, Nevada, USA.
[23] Shinuk Woo, Hwangnam Kim, An empirical interference modeling for link reliability assessment in wireless networks, IEEE/ACM Trans. Netw. 21 (1) (2013).
[24] Muhammad Naveed Aman, Biplab Sikdar, Distinguishing between channel errors and collisions in IEEE 802.11, in: The 46th Conference on Information Sciences and Systems (CISS) 2012, 21–23 March, 2012, Princeton, New Jersey, USA.
[25] Parag Kulkarni, Benjamin Motz, Tim Lewis, Sadia Quadri, Inferring loss causes to improve link rate adaptation in wireless networks, in: The IEEE International Conference on Advanced Information Networking and Applications (AINA) 2011, 22–25 March, 2011, Biopolis, Singapore.
[26] S. Khan, S.A. Mahmud, H. Noureddine, H.S. Al-Raweshidy, Rate-adaptation for multi-rate IEEE 802.11 WLANs using mutual feedback between transmitter and receiver, in: the 21st IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC) 2010, 26–30 September, 2010, Instanbul, Turkey.
[27] Ioannis Pefkianakis, Yun Hu, Songwu Lu, History-aware rate adaptation in 802.11 wireless networks, in: IEEE Symposium on Computers and Communications (ISCC) 2011, 28 June–1 July, 2011, Kerkyra, Greece.
[28] Prashanth Acharya, Ashish Sharma, Elizabeth Belding, Kevin Almeroth, Konstantina Papagiannaki, Rate adaptation in congested wireless networks through real-time measurements, IEEE Trans. Mobile Comput. 9 (11) (2010).
[29] Minstrel, <http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt>.
[30] Kaidi D. Huang, Ken R. Duffy, David Malone, H-RCA: 802.11 collision-aware rate control, IEEE/ACM Trans. Netw. 21 (4) (2013) 1021–1034.
[31] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, Daji Qiao, CARA: collision-aware rate adaptation for IEEE 802.11 WLANs, in: The 25th IEEE INFOCOM 2006, 23–29 April, 2009, Barcelona, Spain.
[32] Shravan Rayanchu, Arunesh Mishra, Dheeraj Agrawal, Sharad Saha, Suman Banerjee, Diagnosing wireless packet losses in 802.11: separating collision from weak signal, in: The 27th IEEE INFOCOM 2008, 13–18 April, 2008, Phoenix, Arizona, USA.
[33] Mythili Vutukuru, Hari Balakrishnan, Kyle Jamieson, Cross-layer wireless bit rate adaptation, in: ACM SIGCOMM 2009, 17–21 August, 2009, Barcelona, Spain.
[34] Daniel Halperin, Wenjun Hu, Anmol Shethy, David Wetherall, Predictable 802.11 packet delivery from wireless channel measurements, in: ACM SIGCOMM 2010, August 30–September 3, 2010, New Delhi, India.
[35] Yeonchul Shin, Munhwan Choi, Jonghoe Koo, Young-Doo Kim, Jong-Tae Ihm, Sunghyun Choi, Empirical analysis of video multicast over WiFi, in: the 3rd International Conference on Ubiquitous and Future Networks (ICUFN) 2011, 15–17 June, 2011, Dalian, China.
[36] Yong He, Beijing, Ruixi Yuan, Xiaojun Ma, Jun Li, The IEEE 802.11 power saving mechanism: an experimental study, in: The IEEE Wireless Communications and Networking Conference, (WCNC) 2008, March 31–April 3, 2008, Las Vegas, Nevada, USA.
[37] Intel website: <http://www.intel.com/support/wireless/wlan/sb/CS-032513.htm>.
[38] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Robust Audio Video Streaming, IEEE std 802.11aa, May, 2012.
[39] John Regehr, Usit Duongsaa, Preventing interrupt overload, in: The ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES) 2005, 15–17 June, 2005, Chicago, Illinois, USA.
[40] Gary W. Scheer, David J. Dolezilek, Comparing the reliability of ethernet network topologies in substation control and monitoring networks, in: Western Power Delivery Automation Conference, April 4–6, Spokane, Washington, 2000.
[41] Youngsam Park, Yongho Seok, Nakjung Choi, Yanghee Choi, Jean-Marie Bonnin, Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LANs, in: The 3rd IEEE Consumer Communications and Networking Conference (CCNC) 2006. 8–10 January 2006, Las Vegas, Nevada, USA.

Dr. Yousri Daldoul received the Engineering degree in Telecommunications from the Higher School of Communication of Tunis (SUP'COM), Tunisia, in 2008. He received the M.Sc. and the Ph.D. degrees in Computer Science from Bordeaux I University, France, in 2009 and 2013, respectively. From 2009 to 2013, He was with Orange Labs networks in, Lannion, France, where he extensively investigated reliability issue related to unicast and multicast stream transport in WiFi and mobile networks. His research field includes Wireless networks, QoS of multimedia applications and protocols optimization.

Djamal-Eddine Meddour received his computer engineering degree with honors from the INI (Institut National d'Informatique), Algiers, Algeria, in 2000, his master's degree in computer science from the University of Versailles, France, in 2001 and his Ph.D. in Computer Science from University of Paris VI in 2004. He is currently a senior researcher with Orange Labs Networks (formerly France Telecom R&D), Lannion, France. His main research activities concern the design and management for light wireless and mobile infrastructure networking, management, and interoperability in new-generation wireless networks. He served as a guest editor and TPC member for numerous international conferences and journals. He is the co-author of several international journal articles and book chapters, and holds many patents.

Toufik Ahmed is a Professor at IPB (Institut Polytechnique de Bordeaux) in the ENSEIRB-MATMECA school of engineering. He is doing his research activities in CNRS LaBRI Lab, UMR 5800 at University of Bordeaux 1. He received the B.Sc. in Computer Engineering, with high honors, from the I.N.I (National Institute of Computer Science) Algiers, Algeria, in 1999, an M.Sc. and a Ph.D degrees in Computer Science from the University of Versailles, France, in 2000 and 2003, respectively. In November 2008, he obtained the HDR degree (Habilitation Diriger des Recherches) at University of Bordeaux-1 on Adaptive Streaming and Control of Video Quality of Service over Wired/Wireless IP Networks and P2P Architectures. His main research activities concern Quality of Service for multimedia wired and wireless networks, end-to-end signaling protocols, P2P network and Wireless Sensors Network.

Raouf Boutaba received the M.Sc. and Ph.D. degrees in computer science from the University Pierre & Marie Curie, Paris, in 1990 and 1994, respectively. He is currently a professor of computer science at the University of Waterloo and a distinguished visiting professor at the division of IT convergence engineering at POSTECH. His research interests include network, resource and service management in wired and wireless networks. He is the founding editor in chief of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (2007–2010) and on the editorial boards of other journals. He has received several best paper awards and other recognitions such as the Premier's Research Excellence Award, the IEEE Hal Sobol Award in 2007, the Fred W. Ellersick Prize in 2008, and the Joe LociCero and the Dan Stokesbury awards in 2009. He is a fellow of the IEEE.